



Sustentado por:

Angel Steven González Fernandez

Numero:

#17

Curso:

4to (E) de Secundaria informática

Facilitador:

Jesús Manuel Fernandez Burgos

Asignatura:

MF_056_3 (Base de datos)

Tema:

Componentes del SQL

Fecha:

23/2/2021

Componentes de SQL



El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Existen tres tipos de comandos SQL:

Los **DLL(Data Definition Language)** que permiten crear y definir nuevas bases de datos, campos e índices.

Los **DML(Data Manipulation Language)** que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos. Los **DCL(Data Control Language)** que se encargan de definir los permisos sobre los datos

DLL (Data Definición Language)

El lenguaje de definición de datos es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen cuatro operaciones básicas: CREATE, ALTER, DROP y TRUNCATE.

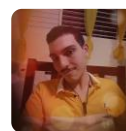
Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

CREATE

Este comando crea un objeto dentro del gestor de base de datos. Puede ser una base de datos, tabla, índice, procedimiento almacenado o vista.

Ejemplo (crear una tabla):

```
# CREATE TABLE Empleado
(
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  Nombre VARCHAR(50),
  Apellido VARCHAR(50),
  Direccion VARCHAR(255),
  Ciudad VARCHAR(60),
  Telefono VARCHAR(15),
  Peso VARCHAR (5),
  Edad (2),
  Actividad Específica (100),
  idCargo INT
)
```



ALTER

Este comando permite modificar la estructura de un objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

Ejemplo (agregar columna a una tabla):

```
# ALTER TABLE 'NOMBRE_TABLA' ADD NUEVO_CAMPO INT;  
# ALTER TABLE 'NOMBRE_TABLA' DROP COLUMN NOMBRE_COLUMNNA;
```

DROP

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

Ejemplo:

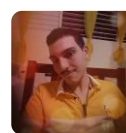
```
# DROP TABLE 'NOMBRE_TABLA';  
# DROP SCHEMA 'ESQUEMA';  
# DROP DATABASE 'BASEDATOS';
```

TRUNCATE

Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

Ejemplo:

```
# TRUNCATE TABLE 'NOMBRE_TABLA';
```



❑ DML (Data Manipulation Language)

Un lenguaje de manipulación de datos es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.
DELETE	Utilizado para eliminar registros de una tabla

INSERT

Una sentencia INSERT de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Forma básica:

```
# INSERT INTO 'tabla' ('columna1', ['columna2,... ']) VALUES ('valor1', ['valor2,...'])
```

Las cantidades de columnas y valores deben ser iguales. Si una columna no se especifica, le será asignado el valor por omisión. Los valores especificados (o implícitos) por la sentencia INSERT deberán satisfacer todas las restricciones aplicables. Si ocurre un error de sintaxis o si alguna de las restricciones es violada, no se agrega la fila y se devuelve un error.

Ejemplo:

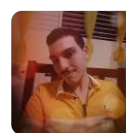
```
# INSERT INTO agenda_telefonica (nombre, numero) VALUES ('Roberto Jeldrez', 4886850);
```

Cuando se especifican todos los valores de una tabla, se puede utilizar la sentencia acortada:

```
# INSERT INTO 'VALUES ('valor1', ['valor2,...'])
```

Ejemplo (asumiendo que 'nombre' y 'número' son las únicas columnas de la tabla 'agenda_telefonica'):

```
# INSERT INTO agenda_telefonica VALUES ('Jhonny Aguiar', 080473968);
```



UPDATE

Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

Ejemplo:

```
# UPDATE mi_tabla SET campo1 = 'nuevo valor campo1' WHERE campo2 = 'N';
```

DELETE

Una sentencia DELETE de SQL borra uno o más registros existentes en una tabla.

Forma básica:

```
# DELETE FROM 'tabla' WHERE 'columna1' = 'valor1'
```

Ejemplo:

```
# DELETE FROM My_table WHERE field2 = 'N';
```

□ DCL (Data Manipulation Language)



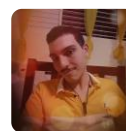
Este es el que consta de declaraciones que controlan la seguridad y el acceso concurrente a los datos de la tabla.

COMMIT

Indica al servidor XDB que haga permanentes todos los cambios de datos resultantes de las declaraciones DML ejecutadas por una transacción.

CONNECT

Conecta el proceso de aplicación (o usuario) a un servidor XDB designado o una ubicación DB2. Esta ubicación se convierte en la ubicación actual para el proceso de solicitud o el usuario.



GRANT (privilegios de base de datos)

Asigna privilegios de acceso a usuarios o aplicaciones de XDB Server.

GRANT (privilegios de secuencia)

Otorga privilegios en una secuencia definida por el usuario.

LOCK TABLE

Extiende las funciones de bloqueo automático de nivel de tabla y registro de XDB Server (en un sistema multiusuario) adquiriendo bloqueos explícitos en una tabla en particular.

REVOKE (privilegios de base de datos)

Cancela los privilegios de acceso para los usuarios o las aplicaciones del servidor XDB.

REVOKE (privilegios de secuencia)

Revoca los privilegios en una secuencia definida por el usuario.

ROLLBACK

Indica al servidor XDB que invierta el efecto de cualquier comando DML ejecutado en una base de datos por una transacción. La información registrada en un registro hacia atrás se utiliza para restaurar la base de datos a un estado existente antes de la transacción.

UNLOCK TABLE

Elimina los bloqueos de la tabla (en un sistema multiusuario) adquiridos con el comando LOCK TABLE.

Las partes suplantada estas vinieran siendo las principales creadas y pasando el tiempo se le fueron agregando mas

Estas como ejemplo serian en si como atributos de mandato para:

CREATE	SELECT
ALTER	INSERT
DROP	UPDATE
TRUNCATE	DELETE
COMMENT	MERGE
RENAME	CALL
	EXPLAIN PLAN
	LOCK TABLE

Está dándonos nuestra serie de funciones que solucionamos por medio de códigos

